

Application of Self-Organizing Techniques for the Distribution of Heterogeneous Multi-Tasks in Multi-Robot Systems

Yadira Quinonez^{""6}, Dario Maravall["] and Javier de Lope["]

^a Computational Cognitive Robotics Group, Dept. Artificial Intelligence,
Universidad Politecnica de Madrid, Madrid, Spain.

^b Facultad de Informatica, Universidad Autonoma de Sinaloa,
Mazatlan, Sinaloa, Mexico.

ay.quinonez@alumnos.upm.es, dmaravall@fi.upm.es, javier.delope@upm.es

Abstract

This paper focuses on the general problem of coordinating of multi-robot systems, more specifically, it addresses the self-election of heterogeneous and specialized tasks by autonomous robots. In this regard, it has proposed experimenting with two different techniques based chiefly on self-organization and emergence biologically inspired, by applying response threshold models as well as ant colony optimization. Under this approach it can speak of multi-tasks selection instead of multi-tasks allocation, that means, as the agents or robots select the tasks instead of being assigned a task by a central controller. The key element in these algorithms is the estimation of the stimuli and the adaptive update of the thresholds. This means that each robot performs this estimate locally depending on the load or the number of pending tasks to be performed. It has evaluated the robustness of the algorithms, perturbing the number of pending loads to simulate the robot's error in estimating the real number of pending tasks and also the dynamic generation of loads through time. The paper ends with a critical discussion of experimental results.

1. Introduction

Autonomous multi-robot systems is an outstanding applied area of Artificial Intelligence that has witnessed a remarkable growth since its inception and that has developed significant progress in several applications [1]. In recent decades, there has been an increasing interest in systems comprised of several autonomous mobile robots, and as a result, there has been a large amount of research done with respect to autonomous mobile robots related to the coordination between them [2, 3, 4, 5]; these investigations have been directed toward finding efficient and robust methods

for controlling these groups of mobile robots. In multi-robot systems (MRS) can often deal with tasks that are difficult, if not impossible, to be accomplished by a single robot.

In the context of MRS, one of the main challenges is the need to control, coordinate and synchronize the operation of multiple robots to perform a specific task. This requires the development of new strategies and methods which allow obtain the desired system behavior in a formal and concise way. More specifically, within MRS, optimal task/job allocation or assignment is an active research problem, in which several global or central allocation methods have been proposed so far [6, 7]. Some authors have also introduced autonomous or decentralized solutions, in particular inspired in the social labor observed in some species of social insects [8,9].

This paper focuses on study the coordination of multi-robot systems, in particular, it addresses the problem of the distribution of heterogeneous multi-tasks in a robust and efficient manner. The main interest in these systems is to understand how from simple rules inspired by the division of labor in social insects, a group of robots can perform tasks in an organized and coordinated way. We take into account a specifically distributed or decentralized approach as we are particularly interested in experimenting with truly autonomous and decentralized techniques in which the robots themselves are responsible of choosing a particular task in an autonomous and individual manner.

2 Formal definitions

2.1 Formal description of the problem

The optimal multi-task allocation problem in multi-robot systems can be formally defined as follows: "Given a robot team formed by N heterogeneous robots, and given K different types of heterogeneous specialized tasks or equiv-

alently, given K different robots roles or robots jobs and given a particular time-dependent load or number of tasks to be executed $L = \{hit, hit, \dots, licit\}$ obtain an optimal distribution of the K tasks among the N robots in such a way that the robots themselves, autonomously and in an individual manner, select a particular task such that all the existing tasks are optimally executed".

Let $L = \{hit, hit, \dots, Ixit\}$ be the different specialized tasks. Each $lj \in L$ has a number of j sub-tasks or pending loads. Let $r = \{r_1, r_2, \dots, r_{jv}\}$ be the set of heterogeneous mobile robots. To solve the problem, we have supposed that all members $R = \{r_1, r_2, \dots, r_{jv}\}$ are able to participate in any sub-task lj .

2.2 Experimental scenario

We have established the following experimental scenario (Fig. 1) in order to analyze a particular strategy or solution for the coordination of multi-robot systems as regards the optimal distribution of the existing tasks. Given a set of N heterogeneous mobile robots in a region, achieving an optimal distribution for different types of tasks. The set of N robots will form sub-teams for each type of task lj . The sub-teams are dynamic over time, i.e. the same robots will not be always part of the same sub-team, but the components of each sub-team can vary depending on the situation.

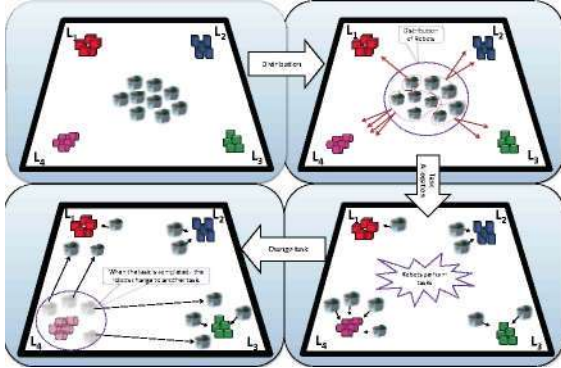


Figure 1. Experimental scenario

Most of the proposed solutions in the technical literature are of a centralized nature, in the sense that an external controller is in charge of distributing the tasks among the robots by means of conventional optimization methods and based on global information about the system state [10]. However, we are mainly interested in truly decentralized solutions in which the robots themselves, autonomously and in an individual and local manner, select a particular task so that all the tasks are optimally distributed and executed. In this regard, we have experimented with response threshold

models and ant colony optimization based on deterministic algorithms to tackle this hard self-coordination problem as described in the sequel.

3 Response threshold models

3.1 A brief introduction

Insect societies are characterized by the division of labor, communication between individuals and the ability to solve complex problems [11], and these characteristics have long been a source of inspiration and subject of numerous studies, acquiring great relevance for many researchers both in the field of robotics as in biology. On the one hand, the biologists trying to prove their theories of social insects on robots, and on the other hand, researchers in the discipline of robotics seek solutions to problems that cannot be solved by a single robot.

Seeley et al. [12] have considered the following experiment to study the collective behavior in a colony of insects, focusing on the work performed by bees to get honey. Two food sources are presented to the colony at 8:00 A.M. at the same distance from the hive: source A is characterized by a sugar concentration of 1.0 mol/l and source B by a concentration of 2.5 mol/l. Between 8:00 A.M. and noon, source A has been visited 12 times and source B, 91 times. At noon, the sources are modified: source A is now characterized by a sugar concentration of 2.5 mol/l and source B 0.75 mol/l. Between noon and 4:00 P.M., source A has been visited 121 times and source B only 10 times. Have shown that a bee has a relatively high probability of going to a good food source and abandon a poor food source.

3.2 Model

Based on these observations, these simple rules of behaviors allow the bees to select the best quality source; Eric Bonabeau et al. have proposed a simple mathematical model of response thresholds for the regulation of division of labor in insect societies [13]. In this model assume that each task is associated with a stimulus or set of stimuli, so that individuals can detect information on each of the different stimulus intensity, therefore, can assess the demand for a particular task when are in contact with the stimulus associated.

Let s be the intensity of a stimulus associated with a particular task; s can be a number of encounters, a chemical concentration, or any quantitative cue sensed by individuals. A response threshold θ , expressed in units of stimulus intensity, is an internal variable that determines the tendency of an individual to response to the stimulus s and perform the associated task. More precisely, θ is such that the probability of response is low for $s < \theta$ and high for $s > \theta$. This

mathematical model that satisfies this requirement is given by:

$$T B_{i l}(s_i) = - \wedge \{ n > l \} \quad (1)$$

where $n > 1$ determines the steepness of the threshold. Fig. 2 shows several such response curves with $n = 2$, for different values of θ . More clearly: for $s < \theta$, the probability of engaging task performance is close to 0, and for $s > \theta$, this probability is close to 1. Then, the probability that an individual will perform a task depends on s .

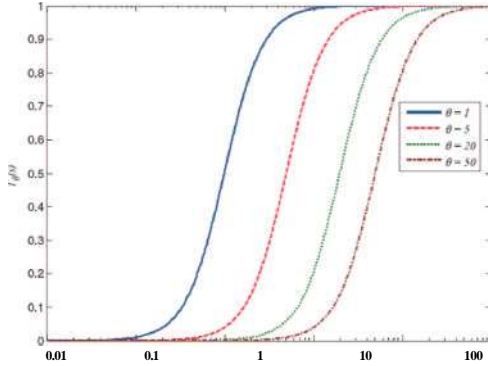


Figure 2. Semi-logarithmic plot with different thresholds ($\theta = 1, 5, 20, 50$) and with $n = 2$.

The underlying idea is very simple, when a stimulus exceeds the threshold of response of an individual, that individual is likely to respond to stimuli, and engage in the task because the level of the stimulus associated with that task exceeds its threshold. The intensity of a stimulus decreases as the individual performs the task; therefore, individuals with high thresholds are unlikely to perform the task when other individuals, with lower thresholds, maintain the stimulus intensity below their thresholds. However, when individuals with low thresholds do not perform the task, individuals that have high thresholds may engage in the task performance because the stimulus intensity exceeds their thresholds.

The tasks can be constant or can be and time-dependent variable. Stimuli associated with each task can vary considerably from one task to another depending on the nature of tasks, task demand and by number of robots that are executing the task. Each task is associated with the demand expressed in the form of a stimulus, when a robot performs a task tends to reduce the intensity of associated stimulus, and as a result, modifies the intensity of the stimuli for tasks that is not running. Each robot $\{r\}$ has a set response thresholds $\theta_r = \{\theta_i, \#2, \dots, \theta_L\}$. Each threshold $\theta_{i,j}$ corresponds to a task type $l_j = \{h, h, \dots, l_j\}$ that the robot is capable of. The initial values of the threshold are randomized to ensure

that their roles are not predetermined; the performance of a given task induces a decrease in threshold of the robots:

$$\theta^{new} = \theta^{old} - a \quad (2)$$

And conversely, the not performance of a given task induces:

$$\theta^{new}_i = \theta^{old}_i + a \quad (3)$$

where $a > 0$ is a factor of increase or decrease that allows to the thresholds vary over time, depending on the performance of tasks.

4 Ant Colony Optimization

4.1 A brief introduction

For over many years, communities or colonies of social insects have been deeply studied by some researchers [8,9], as they provide fascinating examples of functional collective behavior; and are certainly an example of decentralized resolution problems, by the way how these insects perform tasks like finding food, building or expanding their nests, division of labor, etc. In addition, another important feature is that they can solve problems in a way very flexible because it allows adaptation to environmental changes robustly. Therefore, have devoted a great deal of research to figuring out how the social insects achieve these feats. With these researches, have allowed computer scientists to design a variety of "ant algorithms", all of which attempt to capture some amazing qualities of social insects such as self-organization, flexibility, and robustness.

Ant Colony Optimization (ACO) is a meta-heuristic approach that was introduced in the early 1990's by Marco Dorigo in [14, 15]. Since its introduction to the present, a growing number of researchers have been involved in further developing it. The general idea of ACO approach is to solve combinatorial optimization problems based by the behavior of real ants, more specifically, the inspiring source is how ants can find shortest paths between food sources their nest [16]. ACO algorithms are stochastic search procedures based on a colony of artificial ants (computational agents) that work cooperatively and communicate through artificial pheromone trails [17], by means a parameterized probabilistic model [18] called by the authors "the pheromone model".

4.2 Stochastic reinforcement algorithms based reward and penalty

Let's suppose that there are K different specialized tasks, then we designate by $f_{ij}(t)$, the probability at instant t that robot ti selects task l_j these probabilities hold:

$$0 < \frac{1}{N} \sum_{i=1}^N \frac{1}{y_i(t)} < 1; X(t) = \{ \dots \} \quad W$$

where $i = 1, 2, \dots, N$ AT robots and $j = 1, 2, \dots, K$ tasks

Initially, without previous robot's experience these probabilities are initialized at the "indifference" position as follows:

$$y_i(0) = \frac{1}{K} \quad (5)$$

where $i = 1, 2, \dots, N$ AT robots and $j = 1, 2, \dots, K$ tasks

Let us suppose that the action chosen by a generic robot n selects the tasks in a deterministic way based on "forces" $f_{ij}(t)$. Afterwards it starts the learning process, these forces are updated according to the following conventional updating rule:

$$f_{ij}(t+1) = p f_{ij}(t) + (1-p) \frac{1}{N} \sum_{i=1}^N \frac{1}{y_i(t)} \quad (6)$$

where p is the usual learning rate with a fixed value of 0.2; and $\frac{1}{N} \sum_{i=1}^N \frac{1}{y_i(t)}$ is the reward/penalty signal generated by the environment at instant t with the following interpretation: $\frac{1}{N} \sum_{i=1}^N \frac{1}{y_i(t)} = 1$; reward if and only if for the corresponding task j at instant t it holds that $\#i?j(t) < \#L_j(t)$, i.e. the number of robots performing task j is lower than the number of tasks j to be executed; $\frac{1}{N} \sum_{i=1}^N \frac{1}{y_i(t)} = 0$; penalty if and only if $\#i?j(t) > \#L_j(t)$; i.e. when the number of robots performing task j is greater than the number of tasks j or whenever there are not pending tasks to be executed the automata receives a penalty signal. In few words: when the response generated by environment is 1 means that the action is "favorable" and if the response value is 0 corresponds to an "unfavorable" as follow:

$$\frac{1}{N} \sum_{i=1}^N \frac{1}{y_i(t)} = \begin{cases} 1 & \text{If } \#i?j(t) < \#L_j(t) \text{ then reward} \\ 0 & \text{If } \#i?j(t) > \#L_j(t) \text{ then penalty} \end{cases} \quad P = \frac{1}{N} \sum_{i=1}^N \frac{1}{y_i(t)}$$

5 Experimental results

We have conducted several experiments to evaluate the system performance index by applying response threshold models as well as ant colony optimization-based deterministic algorithms to solve the optimal distribution of the tasks among the N robots; so that all of them are executed by means of the minimum number of robots. The ideal objective is that the performance index or learning curve corresponding to the load $l_j(t)$ of each task tend asymptotically to zero for all curves in the minimum time and using the minimal possible number of robots for task execution.

In the simulations we have considered some variants such as: the multi-robot system size, different loads $l_j(t)$ for each type of task, two different ways to carry out the tasks selection, the additive noise generation to simulate the robot's error and the dynamic generation of tasks $l_j(t)$ over

time. According to the results obtained with eq. 1 and eq. 6 we have employed for response threshold models and for the learning automata-based probabilistic algorithms two different mechanisms for the selection of tasks:

1. Maximum principle: at each instant t choose the task that has the highest probability for all $Te_{ij}(sj)$ and $hi(t)$.
2. The strictly random method: using the probabilities $Te_{ij}(sj)$ and $hj(t)$ in the strict sense of the word, it generates a random number with uniform distribution (0 - 1) and it selects the appropriate task to the value obtained by the method of inversion of discrete probability distributions.

5.1 Evaluation of the performance index: by noise or error estimation

To evaluate the evolution of the performance index we have introduced additive noise, perturbing the number of pending loads to simulate the robot's error in estimating the real number of pending tasks. The noise generated is modeled using a normal distribution ("White Noise") as follows:

$$Noise = R + R \cdot S = R(l + S) \quad (8)$$

where $Noise$ is the noise generated to the number of pending loads $k(t)$, which is proportional to the amplitude of the noise R without perturbing, S is a Gaussian distribution with a mean of '0' and a typical deviation '0.005' AT(0,0.005).

Fig.3 and Fig.4 show the evolution of the system performance index obtained for self-election of heterogeneous specialized tasks through response threshold models as well as ant colony optimization-based deterministic algorithms, using both mechanisms: maximum principle and the strictly random method, with a team of robots formed by 20 - 30 heterogeneous robots and 4 types of heterogeneous specialized tasks with different loads. Each experiment has been run 10 times and the results shown are the mean of all.

Fig.3 shows the performance index through threshold response models for the two task selection mechanisms mentioned above and for different values of noise, it can be noted that in all cases the generation of additive noise does not affect the performance of the approach, on the contrary, in most cases better results are obtained with the generation of noise.

Similarly, Fig.4 shows the performance index using ant colony optimization-based deterministic algorithms for both mechanisms and for different values of noise, it can be observed that learning curves corresponding to the load $l_j(t)$ of each task tend asymptotically to zero for both methods. However, when it introduced additive noise in this approach can be clearly seen that in some cases more time is required for the execution of tasks.

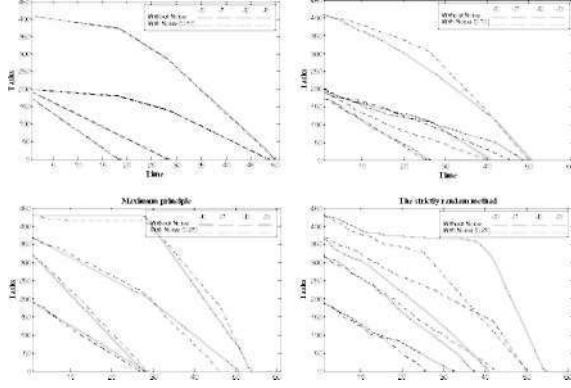


Figure 3. Learning curves with the evolution of the system performance index for self-selection of tasks using Response Threshold Models for different values of noise

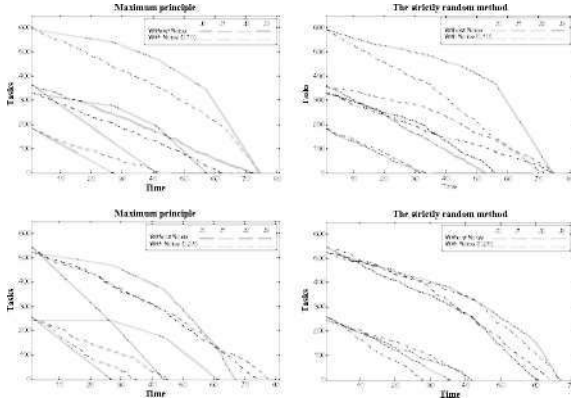


Figure 4. Learning curves with the evolution of the system performance index for self-selection of tasks using Ant Colony Optimization-based deterministic algorithms

5.2 Evaluation of the performance index: by dynamic tasks generation

In the previous experiments, the number of loads for each type of task is determined from the beginning of the simulation and there is not any change until the end of the execution. To evaluate the performance of the algorithm we have generated dynamic tasks. This idea was rescued from classical models of queues simulation, so we have used Poisson distribution to determine the probability of generating a number of tasks through time:

Specifically we will have a different distribution for $k = 1$ to 100. Each A is a positive real number that representing the number of tasks expected to be generated during a time interval. For that expected number of tasks generated is decreasing, and therefore the system is stable, we have parameterized this constant A as follows:

$$A(t) = a - a * t \quad (10)$$

where a is the initial value (for example, 10 or 20) and a is a factor of "reduction tasks" that initially we have defined to 1. Finally, t corresponds the time of execution at each instant.

Fig.5 and Fig.6 show the evolution of the system performance index with dynamic tasks generation through time using the Poisson distribution. Experiments have been performed 10 times and the results shown are the mean of all, we have also additive noise generated in the loads with the maximum principle and the strictly random method. In the results it can be observed dynamic tasks generation, the tasks number generated is decreasing over time. All learning curves tend to zero in both mechanism and not affected the performance for any approaches.

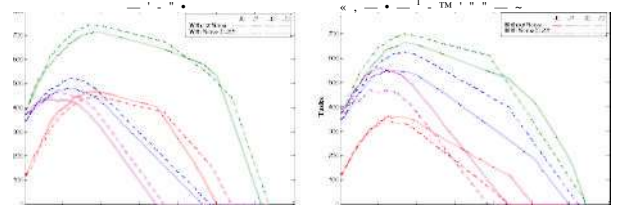


Figure 5. Dynamic tasks generation: learning curves with the evolution of the system performance index for self-selection of tasks using Response Threshold Models

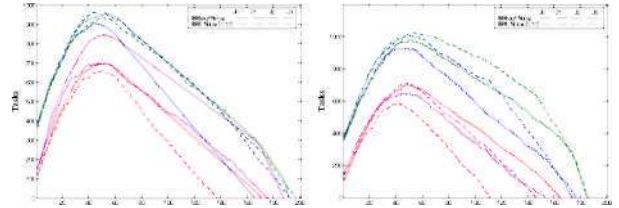


Figure 6. Dynamic tasks generation: learning curves with the evolution of the system performance index using Ant Colony Optimization-based deterministic algorithms

6. Conclusions and further research work

In this paper we have proposed an experimental scenario in order to analyze the self-coordination problem in multi-robot systems, we have compared two different approaches presenting a bio-inspired solution by applying response threshold models as well as ant colony optimization-based deterministic algorithms to solve the problem corresponding to the distribution multi-tasks. More specifically, it addresses the self-election of heterogeneous and specialized tasks by autonomous robots, as opposed to the usual multi-tasks allocation problem in multi-robot systems in which an external controller distributes the existing tasks among the individual robots.

We have analyzed the robustness of each method as regards the estimation error or noise as it is an important and critical parameter concerning the practical viability of these method in real multi-robots scenarios. We have perturbed the number of pending load to simulate the robot's error in estimating the real number of pending tasks and we have also studied the performance index with dynamic generation of loads through time. According to the results obtained the noise generated does not affect the performance of the approaches since the best result are obtained by generating noise in the pending loads. We have shown that both approaches can be efficiently applied to solve this self-coordination problem in multi-robot systems obtaining truly decentralized solutions

References

- [1] L.E. Parker, *"Multiple Mobile Robot Systems"*, In: Bruno, S., Oussama, K. (eds.) Springer Handbook of Robotics, 2008.
- [2] M. Arcak, "Passivity as a design tool for group coordination, In IEEE Transactions on Automatic Control, 52(8): 1380-1390, 2007.
- [3] W Burgard, M. Moors, C. Stachniss, F. Schneider, "Coordinated multi-robot exploration, IEEE Transactions on Robotics, 21(3):376-386, 2005.
- [4] L. Chaimowicz, B. Grocholsky, J.F. Keller, V. Kumar, C.J. Taylor, "Experiments in multirobot air-ground coordination, In IEEE International Conference on Robotics and Automation, volume 4, pages 4053-4058, 2004.
- [5] A. Howard, L.E. Parker, G.S. Sukhatme, "Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection, The International Journal of Robotics Research, 25(5-6):431-447, 2006.
- [6] B.P Gerkey, M.J. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems, International Journal of Robotics Research, 23(9):939-954, 2004.
- [7] R. Farinelli, L. Iocchi, D. Nardi, "Multirobot systems: A classification focused on coordination, IEEE Transactions on Systems, Man, and Cybernetics, Part B, 34(5):2015-2028, 2004.
- [8] G. Oster, E. Wilson, "Caste and ecology in the social insects, Monographs in Population Biology Princeton Univ. Press, 1978.
- [9] G. Robinson, "Regulation of division of labor in insect societies, Annual Review of Entomology, 37(1):637-665, 1992.
- [10] B.P. Gerkey, M.J. Mataric, "Multi-robot task allocation: analyzing the complexity and optimality of key architectures, In IEEE International Conference on Robotics and Automation, volume 3, pages 3862-3868, 2003.
- [11] E. Bonabeau, G. Theraulaz, J. Deneubourg, "Quantitative study of the fixed threshold model for the regulation of division of labour in insects societies, Proceedings Biological Science, pages 1565-1569, 1996.
- [12] T. Seeley, S. Camazine, J. Sneyd, "Collective decision-making in honey bees: how colonies choose among nectar sources Behavioral Ecology and Sociobiology, pages 277-290, 1991.
- [13] E. Bonabeau, G. Theraulaz, J. Deneubourg, "Fixed response thresholds and the regulation of division of labor in insect societies Bulletin of Mathematical Biology, pages 753-807, 1998.
- [14] M. Dorigo, V. Maniezzo, A. Colomi, "The ant system: an autocatalytic optimizing process, Technical Report TR91-016, Politecnico di Milano, 1991.
- [15] M. Dorigo, "Optimization, learning and natural algorithms, Ph.D. thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, 1992.
- [16] C. Blum, "Ant colony optimization: introduction and recent trends, Physics of Life Reviews, 2(4):353-373, 2005.
- [17] M. Dorigo, T. Stutzle, "Ant colony optimization, MIT Press, Cambridge, MA, 2004.
- [18] M. Dorigo, C. Blum, "Ant colony optimization theory: a survey, Theoretical Computer Science, 344(2-3):243-278, 2005.